BIOS56.ASM
8" ONLY
STD. 2.2

```
                ;
                ;       CP/M v2.2 cbios for TEI FDC-2 Controller board
                ;
                ;       CUTER ROM at C000 or E000
                ;       Special disk rom at F000
                ;
                ;
0038 =          msize   equ     56              ;memory in k
F000 =          dskrom  equ     0f000h
9000 =          bias    equ     (msize-20)*1024
C400 =          ccp     equ     3400h+bias
CC06 =          bdos    equ     ccp+806h
DA00 =          bios    equ     ccp+1600h
002C =          nsects  equ     (bios-ccp)/128  ;length to warm boot
0003 =          iobyte  equ     3               ;special port byte
0004 =          cdisk   equ     4               ;current disk
0019 =          souto   equ     19h             ;offset to cuter output
001F =          sinpo   equ     1fh             ;offset to cuter input
0014 =          comport equ     14h             ;comprint printer
0004 =          serport equ     4h              ;serial
                ;
                ;       standard bios transfer vector
                ;
DA00                    org     bios
DA00 C333DA             jmp     boot            ;initial boot
DA03 C36EDA     wboot   jmp     warm            ;warm boot
DA06 C39EDB             jmp     const           ;console status
DA09 C3B3DB             jmp     conin           ;console read
DA0C C3C9DB             jmp     conout          ;console write
DA0F C3E4DB             jmp     list            ;list output
DA12 C3FADB             jmp     punch           ;punch output
DA15 C3F9DB             jmp     reader          ;reader input
DA18 C303DB             jmp     home            ;restore disk
DA1B C3DBDA             jmp     select          ;select disk
DA1E C3F1DA             jmp     settrk          ;set track number
DA21 C3F6DA             jmp     setsec          ;set sector number
DA24 C3FBDA             jmp     setdma          ;set buffer address
DA27 C35ADB             jmp     read            ;read disk
DA2A C379DB             jmp     write           ;write disk
DA2D C3E2DB             jmp     listst          ;return list status
DA30 C398DB             jmp     sectran         ;sector translate
                ;
                ;       disk rom transfer vector
                ;
F003 =          dskboot equ     dskrom+3        ;general boot
F006 =          dskinit equ     dskrom+6        ;init controller
F009 =          dsksel  equ     dskrom+9        ;pysical select
F00C =          dskrest equ     dskrom+12       ;restore to track 0
F00F =          dskseek equ     dskrom+15       ;physical seek
F012 =          dskrd   equ     dskrom+18       ;read sector
F015 =          dskwr   equ     dskrom+21       ;write sector
F018 =          dskrt   equ     dskrom+24       ;read track
F01B =          dskwt   equ     dskrom+27       ;write track (format)
```

```
F01E =          dskrtrk equ     dskrom+30       ;read track #
F021 =          dskwtrk equ     dskrom+33       ;set track #
                ;
                ;       cold boot initialization
                ;
DA33 =          boot    equ     $
DA33 AF                 xra     a               ;init line printer
DA34 D314               out     comport         ;init pia
DA36 3D                 dcr     a
DA37 D315               out     comport+1
DA39 3E3E               mvi     a,3eh
DA3B D314               out     comport
DA3D DB15               in      comport+1       ;clear status
DA3F 3EFF               mvi     a,0ffh          ;send a null
DA41 D315               out     comport+1
DA43 3E36               mvi     a,36h           ;toggle strobe bit
DA45 D314               out     comport
DA47 3E3E               mvi     a,3eh
DA49 D314               out     comport
DA4B 3E81               mvi     a,10000001b     ;initialize iobyte
DA4D 320300             sta     iobyte
DA50 3E00               mvi     a,0             ;clear char buffers
DA52 32B2DB             sta     char
DA55 320400             sta     cdisk
DA58 3A00E0             lda     0e000h          ;look for new roms
DA5B 2F                 cma                     ;ff -> 00
DA5C B7                 ora     a               ;00 ==> old roms
DA5D 3243DC             sta     newrom          ;save for later reference
DA60 CA6BDA             jz      oldroms
DA63 3EE0               mvi     a,0e0h          ;page for new roms
DA65 32FDDB             sta     sout+2          ;change cuter jumps
DA68 3200DC             sta     sinp+2
                oldroms:
DA6B C3B5DA             jmp     gocpm           ;static initialization
                ;
                ;       warm boot - read bdos and ccp
                ;
DA6E =          warm    equ     $
DA6E 310001             lxi     sp,100h         ;reset stack
DA71 CD06F0             call    dskinit         ;reset controller
DA74 0E00               mvi     c,0             ;drive 0 for boot
DA76 CDDBDA             call    select
DA79 CD03DB             call    home
DA7C C26EDA             jnz     warm            ;no go...
DA7F 062C               mvi     b,nsects        ;sectors to read
DA81 0E00               mvi     c,0             ;current track
DA83 1602               mvi     d,2             ;current sector
DA85 2100C4             lxi     h,ccp           ;load address
DA88 =          warm1   equ     $
DA88 C5                 push    b               ;save registers
DA89 D5                 push    d
DA8A E5                 push    h
DA8B 4A                 mov     c,d             ;get sector
DA8C CDF6DA             call    setsec
DA8F C1                 pop     b               ;get dma address
```

```
DA90 C5              push    b               ;save for later
DA91 CDFBDA          call    setdma          ;set read address
DA94 CD5ADB          call    read            ;read it
DA97 C26EDA          jnz     warm            ;error - try again
DA9A E1              pop     h               ;restore dma address
DA9B 118000          lxi     d,128           ;sector length
DA9E 19              dad     d               ;new dma address in hl
DA9F D1              pop     d               ;get sector
DAA0 C1              pop     b               ;restore counter, track
DAA1 05              dcr     b               ;1 less sector to go
DAA2 CAB5DA          jz      gocpm           ;all set
DAA5 14              inr     d               ;next sector
DAA6 7A              mov     a,d
DAA7 FE1B            cpi     27              ;next track?
DAA9 DA88DA          jc      warm1           ;nope
DAAC 1601            mvi     d,1             ;reset sector to 1
DAAE 0C              inr     c               ;increment track
DAAF CDF1DA          call    settrk          ;set track number
DAB2 C388DA          jmp     warm1           ;and continue
                ;
                ;       transfer control to CP/M
                ;
DAB5 =          gocpm   equ     $
DAB5 018000          lxi     b,80h           ;default buffer address
DAB8 CDFBDA          call    setdma
DABB 3EC3            mvi     a,0c3h          ;set up jumps
DABD 320000          sta     0
DAC0 320500          sta     5
DAC3 2103DA          lxi     h,wboot
DAC6 220100          shld    1
DAC9 2106CC          lxi     h,bdos
DACC 220600          shld    6
DACF 3EC9            mvi     a,311q          ;return instruction
DAD1 323800          sta     70q
DAD4 3A0400          lda     cdisk           ;current disk
DAD7 4F              mov     c,a
DAD8 C300C4          jmp     ccp             ;away we go.....
                ;
                ;       set desired drive
                ;
DADB =          select  equ     $
DADB 210000          lxi     h,0             ;error return code
DADE 79              mov     a,c
DADF 323BDC          sta     drvnxt
DAE2 FE01            cpi     1
DAE4 D0              rnc                     ;bad drive
DAE5 6F              mov     l,a             ;*sizeof dp tables
DAE6 2600            mvi     h,0
DAE8 29              dad     h
DAE9 29              dad     h
DAEA 29              dad     h
DAEB 29              dad     h
DAEC 1101DC          lxi     d,dpbase
DAEF 19              dad     d
DAF0 C9              ret
```

```
                        ;
                        ;       set desired track
                        ;
DAF1 =          settrk  equ     $
DAF1 79                 mov     a,c
DAF2 3241DC             sta     track
DAF5 C9                 ret
                        ;
                        ;       set desired sector
                        ;
DAF6 =          setsec  equ     $
DAF6 79                 mov     a,c
DAF7 3242DC             sta     sector
DAFA C9                 ret
                        ;
                        ;       set desired buffer address
                        ;
DAFB =          setdma  equ     $
DAFB E5                 push    h               ;save registers
DAFC 60                 mov     h,b
DAFD 69                 mov     l,c
DAFE 2245DC             shld    dma
DB01 E1                 pop     h               ;restore
DB02 C9                 ret
                        ;
                        ;       send a disk (physically) home
                        ;
DB03 =          home    equ     $
DB03 3A3BDC             lda     drvnxt          ;get desired drive
DB06 B7                 ora     a               ;test
DB07 C0                 rnz                     ;error
DB08 0E00               mvi     c,0             ;set track to 0
DB0A CDF1DA             call    settrk
DB0D CD14DB             call    seldsk          ;select the disk
DB10 CD50DB             call    seek            ;do the seek
DB13 C9                 ret
                        ;
                        ;       select disk for transfer
                        ;
DB14 =          seldsk  equ     $
DB14 C5                 push    b               ;save registers
DB15 3A3ADC             lda     drvcur          ;what was last drive?
DB18 4F                 mov     c,a
DB19 3A3BDC             lda     drvnxt          ;what is desired drive?
DB1C B9                 cmp     c               ;same drive?
DB1D CA48DB             jz      seldsk2         ;yes
DB20 E5                 push    h               ;save register
DB21 213CDC             lxi     h,drvtrk        ;point to track table
DB24 0600               mvi     b,0
DB26 09                 dad     b               ;index by current drive
DB27 CD1EF0             call    dskrtrk         ;get head position
DB2A 77                 mov     m,a             ;save current position
DB2B 3A3BDC             lda     drvnxt          ;get desired drive
DB2E 4F                 mov     c,a
DB2F CD09F0             call    dsksel          ;select it for real
```

```
DB32 213CDC             lxi     h,drutrk        ;point to track table
DB35 09                 dad     b               ;index by drive
DB36 7E                 mov     a,m             ;retrieve its position
DB37 4E                 mov     c,m
DB38 FEFF               cpi     0ffh            ;is it known??
DB3A C243DB             jnz     seldsk1         ;yes it is
DB3D CD0CF0             call    dskrest         ;no - calibrate
DB40 0E00               mvi     c,0             ;set track to 0
DB42 71                 mov     m,c
DB43 =          seldsk1 equ     $
DB43 79                 mov     a,c             ;get position
DB44 CD21F0             call    dskwtrk         ;tell the controller
DB47 E1                 pop     h               ;restore register
DB48 =          seldsk2 equ     $
DB48 3A3BDC             lda     drvnxt          ;set current drive
DB4B 323ADC             sta     drvcur
DB4E C1                 pop     b               ;restore registers
DB4F C9                 ret
                ;
                ;       do a seek
                ;
DB50 =          seek    equ     $
DB50 C5                 push    b               ;save register
DB51 3A41DC             lda     track           ;get desired track
DB54 4F                 mov     c,a
DB55 CD0FF0             call    dskseek         ;do the seek
DB58 C1                 pop     b               ;restore register
DB59 C9                 ret
                ;
                ;       do a read
                ;
DB5A =          read    equ     $
DB5A E5                 push    h               ;save registers
DB5B C5                 push    b
DB5C 3A3BDC             lda     drvnxt          ;get next drive
DB5F B7                 ora     a               ;only 0 is valid
DB60 C276DB             jnz     read1           ;error
DB63 CD14DB             call    seldsk          ;select disk
DB66 CD50DB             call    seek            ;position head, load
DB69 C276DB             jnz     read1           ;error
DB6C 2A45DC             lhld    dma             ;get buffer address
DB6F 3A42DC             lda     sector          ;get desired sector
DB72 4F                 mov     c,a
DB73 CD12F0             call    dskrd           ;do the read
DB76 =          read1   equ     $
DB76 C1                 pop     b               ;restore registers
DB77 E1                 pop     h
DB78 C9                 ret
                ;
                ;       do a write
                ;
DB79 =          write   equ     $
DB79 E5                 push    h               ;save registers
DB7A C5                 push    b
DB7B 3A3BDC             lda     drvnxt          ;get desired drive
```

```
DB7E B7                 ora     a               ;only 0 is valid right now
DB7F C295DB             jnz     write1          ;return with error
DB82 CD14DB             call    seldsk          ;select drive
DB85 CD50DB             call    seek            ;position head, load
DB88 C295DB             jnz     write1          ;error
DB8B 2A45DC             lhld    dma             ;get buffer address
DB8E 3A42DC             lda     sector          ;get desired sector
DB91 4F                 mov     c,a
DB92 CD15F0             call    dskwr           ;write it out
DB95 =          write1  equ     $
DB95 C1                 pop     b               ;restore registers
DB96 E1                 pop     h
DB97 C9                 ret
                ;
                ;       sector translate
                ;
                sectran:
DB98 EB                 xchg
DB99 09                 dad     b
DB9A 6E                 mov     l,m
DB9B 2600               mvi     h,0
DB9D C9                 ret
                ;
                ;       keyboard status
                ;
                const:
DB9E 3AB2DB             lda     char            ;see if char present
DBA1 B7                 ora     a
DBA2 3E00               mvi     a,0             ;set up for cma
DBA4 C2B0DB             jnz     kbdst1          ;already have a char
DBA7 CDFEDB             call    sinp            ;call status/get
DBAA 32B2DB             sta     char            ;save char if any
DBAD 3E00               mvi     a,0             ;clear status
DBAF C8                 rz
                kbdst1:
DBB0 2F                 cma                     ;no char available
DBB1 C9                 ret
DBB2 00         char:   db      0               ;character buffer
                ;
                ;       keyboard input
                ;
                conin:
DBB3 3AB2DB             lda     char            ;get char
DBB6 B7                 ora     a               ;test if present
DBB7 C2C0DB             jnz     kbdin1          ;have char already
DBBA CD9EDB             call    const           ;test for char
DBBD C3B3DB             jmp     conin           ;and try again
                kbdin1:
DBC0 4F                 mov     c,a             ;save char
DBC1 AF                 xra     a               ;clear buffer
DBC2 32B2DB             sta     char
DBC5 79                 mov     a,c             ;restore char
DBC6 E67F               ani     7fh             ;strip parity
DBC8 C9                 ret
                ;
```

```
                ;       video output
                ;
                conout:
DBC9 C5                 push    b               ;save
DBCA 41                 mov     b,c             ;get char
DBCB 3A44DC             lda     lastchar        ;get last output char
DBCE CDDDDB             call    checkcr         ;ignore duplicate cr
DBD1 CADBDB             jz      noprint
DBD4 79                 mov     a,c             ;get char in a
DBD5 3244DC             sta     lastchar
DBD8 CDFBDB             call    sout            ;put it
                noprint:
DBDB C1                 pop     b               ;restore
DBDC C9                 ret
                ;
                ;       routine to check for two cr in a row
                ;
                checkcr:
DBDD FE0D               cpi     0dh             ;was last char a cr
DBDF C0                 rnz                     ;no, print this one
DBE0 B8                 cmp     b               ;is this one a cr too?
DBE1 C9                 ret                     ;return with status
                ;
                ;       list status
                ;
                liststst:
DBE2 AF                 xra     a               ;always 0
DBE3 C9                 ret
                ;
                ;       printer output
                ;
                list:
DBE4 DB14               in      comport         ;get status
DBE6 07                 rlc                     ;test status
DBE7 D2E4DB             jnc     list            ;wait
DBEA DB15               in      comport+1       ;clear status
DBEC 79                 mov     a,c             ;get char
DBED 2F                 cma                     ;invert for comprint
DBEE D315               out     comport+1
DBF0 3E36               mvi     a,36h           ;toggle strobe bit
DBF2 D314               out     comport
DBF4 3E3E               mvi     a,3eh
DBF6 D314               out     comport
DBF8 C9                 ret
                ;
                ;       serial input
                ;
                reader:
DBF9 C9                 ret
                ;
                ;       serial output
                ;
                punch:
DBFA C9                 ret
                ;
```

```
                        ;       cuter interface
                        ;
DBFB C319C0             sout    jmp     0c000h+souto
DBFE C31FC0             sinp    jmp     0c000h+sinpo
                        ;
                        ;       disk parameters
                        ;
                        dpbase:
DC01 11DC0000                   dw      trans8,0
DC05 00000000                   dw      0,0
DC09 47DC2BDC                   dw      dirbf,dpblk8
DC0D E6DCC7DC                   dw      chk00,all00
                        ;
                        ;       8" sector translate table
                        ;
                        trans8:
DC11 01070D13                   db      1,7,13,19
DC15 19050B11                   db      25,5,11,17
DC19 1703090F                   db      23,3,9,15
DC1D 1502080E                   db      21,2,8,14
DC21 141A060C                   db      20,26,6,12
DC25 1218040A                   db      18,24,4,10
DC29 1016                       db      16,22
                        ;
                        ;       8" disk parameter block
                        ;
                        dpblk8:
DC2B 1A00                       dw      26              ;sectors / track
DC2D 03                         db      3               ;block shift
DC2E 07                         db      7               ;block mask
DC2F 00                         db      0               ;null mask
DC30 F200                       dw      242             ;disk size - 1
DC32 3F00                       dw      63              ;directory max
DC34 C0                         db      192             ;alloc 0
DC35 00                         db      0               ;alloc 1
DC36 1000                       dw      16              ;check size
DC38 0200                       dw      2               ;track offset
                        ;
                        ;       scratch area
                        ;
DC3A 05                 drvcur  db      5               ;max drive + 1
DC3B 00                 drvnxt  db      0
DC3C FFFFFFFFFF         drvtrk  db      0ffh,0ffh,0ffh,0ffh,0ffh  ;track table
DC41 00                 track   db      0               ;desired track
DC42 00                 sector  db      0               ;desired sector
DC43 00                 newrom  db      0               ;new rom flag
DC44 00                 lastchar db     0               ;last char put to vdm
DC45 8000               dma     dw      80h             ;buffer address
DC47                    dirbf   ds      128             ;directory scratch
DCC7                    all00   ds      31              ;allocation vector 0
DCE6                    chk00   ds      16              ;check vector 0
DCF6                            end
```